ZJU Summer 2024 Contest 7 By CUHK-Shenzhen

CUHK-Shenzhen

2024 年 7 月 12 日

A. Contest

题意

- 在一次比赛中已知第 1 到 n 个选手通过的题数单调不降，总通过题数为 S。有 m 道题，已知每道题目通过人数在 $[L_i, R_i]$ 之间。
- 对于每个第 i 个人，求他最多通过了多少题。

- -1 当且仅当 $\sum L_i > S$ 或 $\sum R_i < S$ 时发生。
- 对于一名选手 i，我们考虑贪心地让他过的题目尽可能多。
- 因此，我们让编号小于 i 的选手过题是缺乏意义的，而只有让 i 过题可以增加答案，让编号大于 i 的选手过题，可以使得限制更加宽松。我们从这个角度去贪心地构造每道题目的过题选手分布。

A. Contest

- 首先考虑所有 $L \geq n-i+1$ 的题目，我们至少要让 L 个人过这道题，而编号小于 i 的人过题无意义。因此我们直接让编号最大的 L 个人过这道题。每道题对 ans_i 的贡献为 1。
- 接着考虑所有 $L < n-i+1$ 的题目，因为编号小于 i 的人过题无意义，所以我们直接把这些题目的 R 对 $n-i+1$ 取 $\min$。此时，我们只让编号大于等于 i 的选手过题，而目前编号大于等于 i 的选手过题数相等。
- 那么，为了使得 ans_i 最大，我们尽量让每道题都有 R 个人过题，而最优情况下这 $n-i+1$ 位选手的最终过题数量一定是呈现为 $ans_i, ans_i \ldots, ans_i+1, ans_i+1$ 的形式，由于目前 $R \leq n-i+1$，容易证明按照选手编号从大到小循环分配的方式依次分配题目，总能达成这种形式。

A. Contest

- 所以，接下来的这所有 $L < n - i + 1$ 的题目，对 ans_i 的贡献就是把这些题目的 R 对 $n - i + 1$ 取 $\min$ 后，$\min(S, \sum R)/(n - i + 1)$ 下取整。(考虑若是 $S < \sum R$，说明我们总能通过让每道题少取一点的方式来达到 S 的同时下界不低于 L，因为前面 $L < n - i + 1$ 的题目都取的 L 而 $\sum L_i \leq S$; 若是 $S > \sum R$, 我们总能使得所有没到没取到原 R 的题目多取一点来达到 S，因为只要取的是编号最大的若干个选手，选手过题数不降的条件总能满足)。
- 具体维护 $O(n)$ 或者 $O(n \log n)$ 均可。

B. Game

题意

- Alice 和 Bob 玩游戏，轮流每次选择一个进制 p，将 n 写成 p 进制并删除至少 k 个后缀 0.
- 不能操作者输。
- 问先手还是后手获胜。

B. Game

- 注意到题目给的操作相当于选定一个 p，将 n 除以 $p^{k'}$，且 $k' \geq k$.
- 将 n 写成质数乘积的形式：$n = \prod_{i=1}^{s} n_i^{\alpha_i}$，其中每个 n_i 是质数。
- 将 p 写成质数乘积的形式：$p = \prod_{i=1}^{t} p_i^{\beta_i}$，其中每个 p_i 是质数。
- 则 $\beta_i \leq \alpha_i, \beta_i \equiv 0 \pmod{k'}$ 需要成立。
- 可以看作一个从 s 堆大小为 α_i 的石子中轮流取的游戏。每次操作者可以从若干堆中取走一个 $k' \geq k$ 的倍数。不能操作者输。

B. Game

- 因此问题与 p_i 无关，只与可重集 $\{\alpha_i \mid 1 \leq i \leq s\}$ 有关。
- 考虑最后一步不能走的人的局面一定是每堆石子的数量都小于 k. 而任何状态都可以经过取一步到达这个状态：因为操作者可以从每一堆中取走一个 $k' \geq k$ 的倍数，只要取 $k' = k$，就可以令每堆石子的数量对 k 取模，这样一步之后每堆石子数量都小于 k，下一个人无法操作。
- 只要判断存在 $\alpha_i \geq k$，则先手走一步后就后手无法操作。否则先手无法操作。即，先手必胜的充要条件是存在 $\alpha_i \geq k$.

C. Who is the Pig King

题意

- 给定 n 个权值为 a_i 的点，每两个点之间边权为 $a_i \oplus a_j$。
- 求这张图的 MST 的边权。

C. Who is the Pig King

- 考虑模拟 Prim. 注意到边权的值域范围是 $[0, \log V]$. 因此每个点在 Prim 中只会入队 $O(\log V)$ 次。
- 我们只需要精准找出一个点更新之后能松弛的邻居。可以从这个点开始 BFS，每次修改一个 bit。
- 这样每个点只会入队 $O(\log V)$ 次，且图上的总转移数是 $O(V\log V)$（每个点可以修改 $O(\log V)$ 个 bit）。
- 总的时间复杂度是 $O(V\log^2 V)$.

D. Find Palindromic Strings

题意

- 给定一个小写字符串 S。
- 问有个多少个小写回文串长度和字典序都小于等于 S。

D. Find Palindromic Strings

- 考虑回文串为 P，回文串的前半段的长度是否不超过 $LCP(P, S)$.
- 如果不超过，则枚举前半段的长度 l，则前半段即为 $S_{[1,l]}$. 回文串可以由此被唯一确定（要么是 $S_{[1,l]}S_{[l,1]}$，要么是 $S_{[1,l]}S_{[l-1,1]}$）。
- 我们只想快速检查这个回文串的字典序是否不超过 S. 这可以用 Manacher 或者哈希快速找到 $S_{[1,l]}S_{[l,1]}$ 或 $S_{[1,l]}S_{[l-1,1]}$ 和 S 的 LCP 然后判断。
- 因此这一部分的复杂度是 $O(n)$ 或者 $O(n\log n)$.

D. Find Palindromic Strings

- 如果超过，考虑 LCP 的长度为 l，令 a_i 表示第 i 位的字符编号（$a_i \in [1, 26]$）则下一位的选择数量是 $a_{l+1} - 1$。然后考虑回文串的总长度是 $m \le n$，则有 $2(l+1)$ 个位置已经确定。剩下的位置方案数是 $26^{m-2(l+1)}$. 因此总方案数就是：

$$\sum_{l=0}^{(n-2)/2} (a_{l+1} - 1) \sum_{m=2(l+1)}^{n} 26^{m-2(l+1)}$$

$$= \sum_{l=1}^{n/2} (a_l - 1) \frac{26^{n-2l+1} - 1}{26 - 1}$$

- 因此这一部分的复杂度是 $O(n)$ 的。
- 因此总复杂度可以做到 $O(n)$ 或者 $O(n \log n)$.

E. Rectangles

题意

- 给定 $2n$ 个线段 $[l_i, r_i]$ 和 $[s_j, t_j]$（$1 \le i, j \le n$），将 n 个 $[l_i, r_i]$ 与 n 个 $[s_j, t_j]$ 随机形成一个完美匹配（从 $n!$ 种可能中任取一个）。
- 由此形成 n 个矩形，每个矩形的横坐标范围在 $[l_i, r_i]$，纵坐标范围在 $[s_j, t_j]$，面积为 $(r_i - l_i) \times (t_j - s_j)$. 其中 (i, j) 是一对匹配上的线段下标。
- 求 n 个矩形面积并的期望。

E. Rectangles

- 由于每个 1×1 的整点正方形对答案的贡献是独立的。因此我们先考察一个这样的正方形的贡献，也就是它被匹配中的至少一个矩形覆盖的概率。
- 这个正方形 $[x, x+1] \times [y, y+1]$ 被覆盖当且仅当存在一对 (i, j) 匹配上，且 $l_i \le x < r_i, s_j \le y < t_j$.
- 不妨设有 k_1 个 i 满足 $l_i \le x < r_i$，k_2 个 j 满足 $s_j \le y < t_j$. 在这种情况下，至少有一对 (i, j) 匹配上的方案数是 $n! - \dfrac{(n-k_1)!(n-k_2)!}{(n-k_1-k_2)!}$.
- 而 k_1 只和 x 有关，k_2 只和 y 有关。设有 $f(k_1)$ 个 x 被 $[l_i, r_i]$ 覆盖 k_1 次，$g(k_2)$ 个 y 被 $[s_j, t_j]$ 覆盖 k_2 次，则 $f(k_1), g(k_2)$ 可以由离散化加差分再前缀和的方式预处理。

E. Rectangles

- ，则总的答案就是：

$$\frac{1}{n!}\sum_{k_1=1}^{n}\sum_{k_2=1}^{n} f(k_1)g(k_2)\left(n! - \frac{(n-k_1)!(n-k_2!)}{(n-k_1-k_2)!}\right)$$

$$= \left(\sum_{k_1=1}^{n} f(k_1)\right)\left(\sum_{k_2=1}^{n} g(k_2)\right) -$$

$$\frac{1}{n!}\sum_{k=1}^{n}\frac{1}{(n-k)!}[x^k]F(x)G(x)$$

- 其中 $F(x) = \sum_{k_1=1}^{n}(n-k_1)!f(k_1)x^{k_1}$, $G(x) = \sum_{k_2=1}^{n}(n-k_2)!g(k_2)x^{k_2}$.
- 这是一个可以用 NTT 在 $O(n\log n)$ 时间内解决的问题。

F. Inevitable Checkmate

题意

- 给定国际象棋中的王/后/车/象/马，q 次移动，每次移动前后能攻击到几个格子。

F. Equivalent Rewriting

没有移动的情况

- 车显然。
- 王只和有多少个维度贴边界有关。贴边界的维度贡献是 2，否则是 3.
- 后枚举长度之后就是王的情况。
- 象枚举长度之后只和有多少个维度有一个/两个方向有关。从这些方向中任选两个，再减去选的两个是同一个维度的情况。
- 马同样选择一个长度为 1 的方向和一个长度为 2 的方向再减去选到了同一个维度的情况。

F. Equivalent Rewriting

有移动的情况

- 对每个长度 l 用 x_l 表示有一个方向的维度数量，用 y_l 表示有两个方向的维度数量。
- 则我们只需要维护 $\binom{x_l+y_l}{2}$，$2^{x_l}3^{y_l}$，y_l 三个量对 l 求和的结果。
- 当单维度修改的时候，它对每个 x_l, y_l 的修改都至多只有 1，且影响相同的是一个区间。
- 因此用线段树维护即可。时间复杂度 $O(n \log n)$.

Thank you!